ReadMe file for codes in this repository

Vito A R Susca

January 14, 2021

Contents

1 Generation of ER matrices

Here we describe extensively the MatLab algorithm *create_ER_matrix.m*. It allows one to derive quickly the adjacency matrix of an Erdős and Rényi (ER) graphs. The generation of single instances of adjacency matrices of ER graphs is very simple. Indeed, by fixing the size N and the mean degree c, one obtains the link probability $p = c/N$. Then, a logical $N \times N$ $\{0, 1\}$ matrix X is created, according to the following rule:

$$X_{ij} = \begin{cases} 1 & U < p \\ 0 & U \geq p \end{cases} , \tag{1}$$

where a different realisation of $U \sim \mathrm{Unif}[0, 1]$ is drawn for each entry X_{ij}. After the matrix X has been created, just select the upper triangular part T of the matrix excluding the main diagonal, and sum the matrix T to its transpose, yielding $J = T + T^T$. It is easy to check that J is the adjacency matrix of an ER graph with mean degree c and N nodes. This can be checked by comparing the actual degree distribution of J with the expected Poisson degree distribution with parameter c. Further comments can be found in the MatLab file.

2 Algorithm for the solution of the single-instance cavity equations

In this section we describe extensively the MatLab algorithm *Spectra_single_instance.m*. Given a single $N \times N$ sparse symmetric random matrix J, this algorithm allows one to derive the spectral density of J using Eq. (29) and (31) and finally (32). Further comments can be found in the MatLab file.

We start with a set of preliminary tasks.

- Obtain the degree sequence of the graph underlying the matrix J by counting the number of non-zero entries in every row (or column) of J. Store the sequence in an array $\boldsymbol{d}$ of length N.

- Build a table NB with N rows, one for each node. Each row $\mathrm{NB}(i)$ is an array with the label i in its first position, $\mathrm{NB}(i)(1) = i$, and the degree $d(i)$ of i in its second position $\mathrm{NB}(i)(2) = d(i)$. Each of the subsequent entries of $\mathrm{NB}(i)$ are the labels of the neighbours of i. Clearly, the length of each row $\mathrm{NB}(i)$ is $d(i) + 2$.

- Fix a set I of equally spaced real positive numbers, starting at zero. The parameter λ will take values in I. Denote the distance between two consecutive values in I as $\Delta\lambda$. It is convenient to normalise the entries of J by $\sqrt{c}$, where c is the mean degree. (For a weighted adjacency matrix this amounts to asking that the variance of the bond weight distribution scale as $1/c$). In this way, the range I will be typically $[-3.5, 3.5]$. The single-instance spectral density in Eq.(32) is an even function of λ. Therefore, it can be evaluated in the interval I and then simply mirrored w.r.t. 0 to obtain the full spectral density shape.

- Choose the value of the regulariser ε. In order to properly represent delta-like peaks in the spectrum, ε should be choosen of the same order of $\Delta\lambda$ or smaller. Indeed, ε is the width of the Cauchy distribution that in this context approximates numerically delta peaks. If $\Delta\lambda \gg \varepsilon$, a peak may extend across two consecutive data points at which $\rho(\lambda)$ will be computed, hence it will not be detected.

After these preliminary steps, the following series of steps are iterated for each value of $\lambda \in I$.

1. Create a $N \times N$ matrix W, with zero entries. Each matrix entry W_{ij} such that $J_{ij} \neq 0$ will represent a cavity field, i.e. $W_{ij} = \omega_j^{(i)}$ for any i and j that are actually connected. Each entry W_{ij} such that $J_{ij} \neq 0$ is initialised as a complex random number with a positive real part, i.e.

$$W_{ij} = 10 + U_{ij} + \mathrm{i}Z_{ij} \ , \tag{2}$$

   where for any couple of connected nodes (i, j) we independently draw two random numbers $U_{ij} \sim \mathrm{Unif}[0, 1]$ and $Z_{ij} \sim \mathcal{N}(0, 1)$. All other entries are left equal to zero.

2. Select a tolerance $\epsilon_w \leq 10^{-10}$. Define the ratio

$$R = \frac{|W_{\mathrm{curr}} - W_{\mathrm{old}}|}{|W|_{\mathrm{old}}} \ , \tag{3}$$

where $|A|$ denotes the Frobenius norm of the matrix A, i.e. $|A| = \sqrt{\sum_{i,j} A_{ij}^2}$. In our case, only the entries W_{ij} such that $J_{ij} \neq 0$ corresponding to the cavity fields $\omega_j^{(i)}$ contribute to the norm, since all other entries are zero. The ratio R represents an indicator of convergence for the set of recursion equations (29) during iterations. It is the ratio of the norm of the difference between W in the current and the previous state, $|W_{\text{curr}} - W_{\text{old}}|$, and its norm in the previous state, $|W|_{\text{old}}$. Before the very first iteration of the while loop of step (3), compute the value of the variable $|W|_{\text{old}}$ using the initial instance of W defined in step (1). Moreover, the initial value of R is set to 2 by default.

3. While $R > \epsilon_w$, the following routine is performed for $i = 1, ..., N$. When $R < \epsilon_w$, jump to step (6)[1].
   (a) If $d(i) \geq 1$, denote the set of neighbours of i by $n(i)$. $n(i)$ is obtained selecting $d(i)$ consecutive elements from the i-th row of the table NB, starting from the 3^{rd} position: $n(i) = \text{NB}(i)(2+s)$, for $s = 1, ..., d(i)$. If $d(i) = 0$, skip to the next value of i.
   (b) For any $j \in n(i)$, denote the set of neighbours of j except i by $n(j \backslash i)$. As before, $n(j \backslash i)$ is obtained selecting first all the $d(j)$ consecutive elements from the j-th row of the table NB starting from the 3^{rd} position, and then excluding the element equal to i. Hence, first we get $n(j) = \text{NB}(j)(2+s)$ for $s = 1, ..., d(j)$ and then we eliminate the entry of $n(j)$ equal to i, obtaining a set of $d(j) - 1$ node labels.
   (c) Compute
   $$W_{ij} = \mathrm{i}\lambda_\varepsilon + \sum_{\ell \in n(j \backslash i)} \frac{J_{j\ell}^2}{W_{j\ell}} \ . \tag{4}$$
4. After the loop over $i = 1, ..., N$ has been completed, compute the ratio R in (3), using the updated matrix W to compute the value of $|W|_{\text{curr}}$. Then, assign the value of $|W|_{\text{curr}}$ to the variable $|W|_{\text{old}}$.
5. Go to step (3).
6. Create an array $\boldsymbol{w}$ of length N, with zero entries. Each element of this array will represent a single-site inverse variance, i.e. $w_i = \omega_i$. Then, for any $i = 1, ..., N$:
   (a) If $d(i) \geq 1$, consider again the set of neighbours of i, denoted by $n(i)$. Compute
   $$w_i = \mathrm{i}\lambda_\varepsilon + \sum_{j \in n(i)} \frac{J_{ij}^2}{W_{ij}} \ . \tag{5}$$
   (b) Else, if $d(i) = 0$, then set $w_i = \mathrm{i}\lambda_\varepsilon$.

[1]The condition $R < \epsilon_w$ is trivially false when entering the loop in step (3) for the first time.

7. Compute the spectral density (32) as

$$\rho(\lambda) = \frac{1}{\pi N} \sum_{i=1}^{N} \mathrm{Re}\left[\frac{1}{w_i}\right] \ . \tag{6}$$

Step (7) concludes a single interation of the algorithm for a given value of $\lambda \in I$. When $\rho(\lambda)$ is computed for any $\lambda \in I$, the full spectral density of the matrix J is obtained.

3 Population Dynamics for the spectra of ER matrices

The MatLab algorithm *Pop_Dyn_for_spectra_ER.m* allows one to obtain the average spectral density of the ensemble of adjacency matrices of ER graphs. The pseudocode of the population dynamics algorithm is detailed in Section 6 of our paper. Further comments can be found in the MatLab file.